



Implementando as primeiras telas do Projeto Integrado com a biblioteca Material-UI

Chegou a hora de começarmos a implementar as telas de nosso Projeto Integrado. Tal projeto, como mencionado em outro momento, se trata de um CRUD para controle de projetos, tarefas e recursos. Logo, teremos telas responsáveis pela criação, recuperação, edição e deleção de registros. A lógica dessas telas pode variar de sistema para sistema, mas costumam seguir o seguinte fluxo:

1. O usuário realiza seu login na aplicação;

2. O usuário acessa a página home e através de um menu, tem acesso às funcionalidades do sistema;

3. O usuário escolhe uma opção no menu, por exemplo, Projetos;

4. O usuário é levado até a página de Projetos, na qual visualizará uma listagem, em formato tabular, de todos os projetos existentes atualmente;

5. Na tela de listagem, o usuário tem disponíveis os botões para as seguintes ações:

    a. Cadastrar um novo registro (trata-se de um botão global);



b. Editar um registro existente (trata-se de um botão individual, disponível a cada linha de registros);

c. Deletar um registro existente (trata-se, também, de um botão individual, disponível a cada linha de registros).

Além do fluxo básico acima, nosso projeto, dados os relacionamentos existentes entre as entidades, contará ainda com cadastros auxiliares, a partir de cadastros principais. Por exemplo:

- Para cadastrar um projeto será necessário escolher, entre os recursos disponíveis, quem será o seu gerente;
- A escolha do gerente se dará a partir de um componente de lista (select / dropdown);
- Caso o gerente do projeto ainda não tenha sido cadastrado, deverá ser possível cadastrá-lo a partir da tela de cadastro de projetos, para que não seja necessário sair da tela, cadastrar o usuário e depois disso retornar.

Além das funcionalidades e do fluxo descrito acima, devemos ter em mente, ao iniciarmos um projeto que contenha frontend, as questões visuais e de usabilidade do mesmo. Sobre as questões visuais, o React Js fornece componentes próprios que nos permite construir toda a aplicação. Entretanto, há bibliotecas de terceiros, disponíveis, que podem nos auxiliar nessa tarefa. Em nosso projeto, veremos duas dessas bibliotecas: a Material-UI e a Bootstrap. Inicialmente, codificaremos utilizando a Material-UI. Posteriormente, assunto que trataremos em outro módulo, veremos a Bootstrap.

Cabe destacar outro recurso que utilizaremos em nosso projeto e que introduziremos neste módulo: o acesso às APIs utilizando a

biblioteca Axios. Tal funcionalidade será necessária para integrarmos o nosso front com o backend da aplicação.

Agora, vamos falar sobre a Material-UI: como já mencionado, trata-se de uma biblioteca que não pertence ao Core do React Js, mas que pode ser incorporada em projetos Rect com grande facilidade. Tal biblioteca fornece uma suíte de componentes de Interface de Usuário (UI) composta por formulários, botões, ícones, listas, tabelas, elementos de navegação, etc. A lista é bastante extensa e pode ser conferida em detalhes no site oficial. Além disso, também no site oficial, há inúmeros exemplos de utilização desses componentes, onde fragmentos de códigos são disponibilizados. A utilização desta biblioteca é tão simples que basta instalá-la, importá-la em nosso código e usarmos os componentes que quisermos (rola até um copia e cola dos códigos de exemplo, com as devidas edições). Um ponto importante sobre a Material é que estão disponíveis versões free e também versões pagas (com diferentes valores, de acordo com os recursos oferecidos). Entretanto, a versão gratuita atende a maioria dos projetos e, portanto, nos atenderá muitíssimo bem.

O próximo passo, após essa pequena introdução sobre a Material UI, é usá-la em nosso projeto. Como em todos os casos, em que precisaremos instalar bibliotecas adicionais, utilizaremos um gerenciador de pacotes para essa tarefa. Logo abaixo, você pode conferir o comando de instalação com o npm (mas, você pode utilizar também o yarn). Detalhe: crie um novo projeto React, entre na pasta do projeto e, estando nela, execute o comando. O processo completo pode ser visto abaixo:

```
npx create-react-app template-materialui
cd template-materialui
npm install @mui/material @emotion/react @emotion/styled @mui/icons-material
```



Após criado o projeto e instalada a lib MaterialUI, vamos construir o primeiro CRUD de nossa aplicação, o de Tarefas. Aqui, o primeiro passo consiste em criarmos, dentro da pasta “src”, uma pasta para armazenar as páginas de nossa aplicação. Então, crie uma pasta chamada “pages”. Dentro dela, crie uma nova, chamada “tarefa”.

Em uma aplicação é normal termos as páginas e também outros elementos, como componentes que são compartilhados pelas páginas, além de outros recursos, como serviços/chamadas a API, etc. A fim de manter nossa aplicação bem organizada e também facilitar o reaproveitamento de código, além de manter nossos componentes focados em responsabilidades específicas, vamos criar mais uma pasta, chamada “components”. Nessa pasta armazenaremos todos os componentes que serão utilizados pelas páginas de nossa aplicação. Ela deverá ser criada no mesmo nível da pasta “pages”. Nosso primeiro componente será o “header”, responsável por mostrar algumas informações gerais, como o nome da aplicação, e também conter o seu menu – que será criado mais adiante. Depois, importaremos este header em todas as nossas páginas.

A partir desse ponto, já podemos começar a codificação da tela inicial, a que lista as tarefas existentes e fornece os botões de ação para criarmos uma nova tarefa, editarmos e/ou deletarmos as existentes. Você poderá tentar criar essa página sozinho, a partir de algum tutorial ou poderá baixar o código disponível no repositório

abaixo – que está funcional, ou seja, é clonar e rodar. Lembre-se de que um requisito fundamental para sistemas web é sua responsividade, ou seja, que as páginas de nossa aplicação sejam adaptáveis aos dispositivos com diferentes tamanhos de tela. Nesse ponto, temos outra grande vantagem na utilização de bibliotecas como a Material UI, pois ela fornece um sistema de Grid que facilita tal tarefa. Você poderá conferir isso no código, em que o componente <Grid> é utilizado nesse sentido.

OBS: Normalmente, ao clonar um projeto, você precisará instalar as dependências do mesmo, que ficam listadas no arquivo package.json. Para isso, siga esse passo a passo:

1. Clone o projeto a partir da url git;

2. Entre na pasta do projeto, via terminal;

3. Execute o comando: npm install (ou yarn , caso utilize esse gerenciador);

4. Aguarde o término da instalação das dependências;

5. Para rodar o projeto e acessá-lo via navegador, execute: npm start (ou yarn start).

Link: https://github.com/FaculdadeDescomplica/pratica_integradora_tecnologias_disruptivas/tree/main/modulo10/template-materialui **(Acesso em 29/09/2022)**

No código são utilizados vários recursos da biblioteca React JS, vistos ao longo da disciplina. Outro ponto importante nos códigos acima é que foram utilizados dados “fake”, definidos localmente. Essa

implementação serve para fins de demonstração e aprendizagem. Na prática, os dados serão recuperados e persistidos em banco de dados, através de chamadas para o backend com uso da biblioteca Axios (falaremos mais sobre essa biblioteca em outra ocasião). Cabe ressaltar que as partes fundamentais do código foram todas comentadas diretamente no código. Logo, ao ler o código, você entenderá o que está sendo feito em cada parte. Além disso, poderá alterar e modificar a seu critério, a fim de aprimorar seus conhecimentos.

Atividade Extra

A Material-UI é uma biblioteca rica em recursos. Como atividade extra, recomendo que você acesse a documentação oficial, navegando pelos detalhes da lib e também pelos códigos disponibilizados.

Segue o link: https://mui.com/pt/ **(Acesso em 29/09/2022)**

Referência Bibliográfica

MORGAN, Joe. **How to code in React.js**. Digital Ocean, 2021.

Ir para exercício